VuFind development revisited
lessons learned from one year of migration to VuFind2.x
Some background – the finc project

EFRE funded project from 06/2011 – 12/2014 bringing VuFind to 11 libraries in Saxony.
Easy workflow

Each developer had an individual dev-system (lamp, wamp, etc.) committing to subversion which gets deployed manually for initial testing on Alpha.
Simple VuFind-Structure

Each library has its conf, theme and additional ILS-driver in finc-VuFind.

- harder to manage with growing number of libraries
- lots of customization made contribution to community increasingly difficult
Doing it better with VuFind2.x

- first steps for transition in 10/2014
- lots of code that needs to be ported
- main goal: realigning with the community code
- switch from subversion to git
What to keep?

Workflow - make easy workflow easier but more organized!

Not simple structure - create a scalable structure with VuFind2 code-wise & organizational.
Making an easy workflow easier

Standardize development environment
- docker image for VuFind2.x developer
- Debian based lamp stack
- debugging (Xdebug)
- testing (phpunit, phpcs)

https://github.com/finc/docker-vufind2
Making an easy workflow easier

Use git-hooks for deploying

- tags triggering hooks
- deploying Alphas as disposables
- deploying Staging after passing Jenkins
- deploying Live still manual and by non-developer – sharing responsibilities
Restructuring files

Using VuFind2 multisite capabilities

- config-, theme- and lang-files all in institution specific folders
- simple naming schema
  vufind2/isil/
  vufind2/isil/lang
  vufind2/themes/isil
- optional custom module folder
  vufind2/module/isil
Restructuring files

Using inheritance

- 3 layers for config and language files:
  - global*
  - instance**
  - context***

- finc theme as abstraction layer for theme framework (foundation5, bootstrap3)
Workflow + Structure = Structured Workflow?

- extensive usage of branching in git:
  - each issue is a single branch: issue/####
  - each VuFind-instance is a single branch: instance/isil
- devs use dev subfolder for local configuration and development purposes
- deploy-stages use appropriate subfolders for stage-specific configuration
And getting code back to the VF-community?

- features encapsuled in issue branches
- pullrequest to community after finished implementation (pullrequest_####)
- pullrequests against latest vufind-org master
What are the lessons learned?

- automated deploy and docker environment saves lot of time
- config structure works well – reduces complexity (but only works for ini files)
- workflow separates coding tasks more clearly, but adds organizational load on developers side (git juggling)
- workflow to integrate continued work on community pullrequests into local master/issue still to be improved
Thanks!

André Lahmann
lahmann@ub.uni-leipzig.de